Information Awareness on the Desktop: A Case Study

D. Scott McCrickard

Graphics, Visualization, and Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

E-mail: mccricks@cc.gatech.edu

Technical Report GIT-GVU-97-03
January 1997

Abstract

In the future, the amount of online information will increase as more people use the Internet for a growing number of activities. Staying updated on the status of important information requires a tool that will monitor information resources and alert the user when changes occur. To remain both useful and unintrusive, such a tool must provide quick access to information in a small space. To address this need, we created Irwin, a highly configurable information monitoring tool. This paper describes Irwin and examines how four users used it over a five-month period.

1 Introduction

The dynamic nature of the Internet results in a constant flood of new information to users. A variety of tools can organize, filter, and display this information, but each of these tools requires regular accesses to stay updated on important events. If users know when and in what ways the information changes, they can make informed decisions on when to access a resource, thus better utilizing their time. These changes need to be summarized in a way that will maximize comprehension while minimizing effort.

While it is important for a user to obtain information about interesting resources at any time with minimal effort, the information should be provided in a way that does not interrupt the user's work. This combination of awareness and unintrusiveness suggests that a tool should require little screen real estate. Most computer desktops have small tools that monitor the time, the system load, and the email inbox; users may be willing to sacrifice a comparable amount of space to increase their information awareness.

In the future, the information explosion will be magnified as the number of people using the Internet increases. In addition, the increasing popularity of laptop computers, PDAs, and wearable computers will put an increased premium on screen space. The luxuries provided by reduced-space information visualization techniques may become necessities in the near future.

Much of the research in this area focuses on the filtering and dissemination of information [FD92, YGM95]. However, these typically generate text-only output, which would be difficult for users to assimilate in a small space. In this paper we explore techniques for small-space display to examine if and how people will use them in their workplace as part of awareness tools, and if they can be adapted and improved to optimize the use of the available space. Specifically, we introduce Irwin, an awareness tool which uses many of these techniques, and we describe a case study that examines how four people used Irwin in their workplaces over a five-month period.

2 Irwin

Irwin [MR96] monitors Internet information resources and alerts the user of updates and modifications. Irwin consists of a set of hypertools–small reusable programs that can run simultaneously and share information. The central tool in Irwin handles the visualization and user interactions, while the remaining tools process the information from each resource and update the visualization tool when important changes occur.

The information resources monitored by Irwin can include email folders, Usenet newsgroups, Web pages, and weather data. The email and newsgroup hypertools monitor the messages in a folder or group, alerting the user when new messages arrive. The Web tool summarizes headers, lists, and hypertext links on a Web page, allowing the user to monitor news wires and hotlists. The weather tool monitors the weather conditions and forecast for a given city. Modifications are displayed via changes in the visualization.

Irwin uses multiple views to convey an overview of each resource plus details about a selected resource: an icons view, an auditory cue, a navigation bar, and several textual views. In the icons view, the state of each resource is given by an icon. When the resource changes, the appearance of the icon changes and an auditory cue is played. If user selects

Figure 1: Irwin's display of an email folder. The icons reflect the state of each resource and the time since a change occurred. The navigation bar contains a coded line representing each recently received message, indented proportional to the time at which the message was received. The text views provide detailed information about selected messages.

an icon, the other views are updated to show information about the corresponding resource. A navigation bar shows a syntactic encoding of the resource contents and can be used to select the messages displayed in the textual views. Users can configure the orientation and placement of the icons with respect to the other views and can even choose to hide the other views until some event happens, e.g. an icon is clicked.

2.1 Icons

Icons are used in interfaces because they provide a universal representation in a small amount of space. In a WIMP environment they can both provide information and invite the user to click to obtain more information. Studies have shown that changing the appearance of icons can convey additional information about their use; for example, animated icons [BSM91] convey more information about the functionality of tools in a tool palette than a static image. While Irwin does not use animated icons, it does change the color and appearance of icons based on the contents of the information resources they represent. It is our hope that these changes will convey information about the status of the resource.

Irwin uses a 16-by-16 pixel bitmap in its icons to show the status of information resources and to provide a gateway to more detailed views. The color of the icons change to reflect the recency of updates to the resource: the resource's icon is originally black, but when a resource is updated its icon changes to red. Over time, if the resource is not updated its icon color fades. It never fades entirely to black until the user clicks on it.

In addition, we explored the possibility of changing the bitmap itself based on the state of the resource. The weather icon changes its appearance to a sun, a cloud, or rain to reflect the current weather conditions.

2.2 Auditory cues

Auditory cues provide an additional mechanism for alerting users that a change has occurred. They may be the only way the user knows about the change if the display is obscured or if the user is not looking at the screen. Auditory cues have proven useful in both enhancing and replacing visual cues in user interfaces [BNG89, Bre94].

Based on these previous results, we felt it would be useful to allow users to combine an auditory cue with the visual icon change. We selected sounds that are distinctive but are

not distracting and, most importantly, are short. Each is less than two seconds in duration, which we expect will be long enough to be noticed but not too long to become an annoyance. In our initial configuration, we tried to select sounds that were indicative of the resource; for example, a dog bark for mail (dogs always bark at the mail carrier) and a splat for news (similar to a newspaper hitting a front porch). Of course, the user can select any desired sound; we provide a list of twenty.

2.3 Navigation bar

Irwin needs a method for visualizing and navigating a list of document summaries. While a scrollbar is a widely-used navigation tool, it shows little information about the content of the list. Since screen space is at a premium in Irwin, we would like to use the space occupied by a scrollbar not only for navigating the list but also for providing an overview of the contents of the list. To accomplish this, we need a graphical method for reducing a line or word of text into a set of pixels.

One method used in read wear and edit wear [HH94] is to represent the text with a line of pixels. Characteristics like length and color correspond to properties of the text such as the number of modifications. The SeeSoft [ESS92] and RunView [MA96] software visualization programs represent code with lines of pixels. The length of the line can correspond to the length of the line or function, and the color can correspond to the author, module name, or modification date.

Rather than encode semantic information in Irwin's navigation bar, we chose to use a syntactic encoding. The advantage of a syntactic encoding (based only on the structure of the word being encoded) is that it remains consistent between sessions; users who remember an encoding from one session can look for it again in later sessions. The semantic encodings described above require a legend to explain to the user the meanings for each color. Since a legend takes up a lot of space and space is at a premium, we chose to use a syntactic encoding instead.

For email and Usenet news, Irwin represents each message with the author's encoded email address, where a 4-by-4 block of pixels represents each character in the word. To differentiate between words, Irwin colors the blocks that correspond to vowels such that 'a' is red, 'e' is orange, 'i' is yellow, 'o' is green, and 'u' is blue. This method seems to generate different patterns for a large number of email addresses; for 38 senders to a mailing list over a three-week period, only two shared the same encoding.

For Web pages, Irwin uses the same syntactic encoding but must first identify a word to encode by analyzing the structure of the sentence to find the subject. As identifying the best word is a difficult task, we do not expect users to do as well at identifying words. However, users should still be able to pick out repeated patterns.

As a supplement to the syntactic encoding, we allow users to specify their own keyword-to-colors mapping. This method performs a boolean match such that if a word or name appears in the message, the line of pixels will be highlighted with the appropriate color (see Figure 2). We expect that over time some users will turn off this syntactic encoding and use keyword matching only.

We considered using a fisheye view [Fur86] to show all of the list items at once. In a fisheye view, the size and position of the graphical representation are dependent upon

Figure 2: Irwin's display of the ESPNET sports news wire. The navigation bar contains a coded line representing a keyword for each header indented proportional to the time at which the message was received. Certain headers are highlighted - red for baseball, green for tennis, and blue for football. The message view provide detailed information about the selected message.

the location of the focus. This allows both focus and context to be shown simultaneously. However, fisheye views can be very slow to update because a change in focus results in a change in the size and position of a number of items. In addition, when a graphical area is fisheyed, fewer items can be seen at full size because the space is used to show other items at a reduced size. Since space is at a premium, we chose to see more items at full size and scroll when the focus reaches the top or bottom of the display. We expect that users will be more concerned with the most recent messages anyway and will not scroll very often.

We also encode the time at which the message was stamped by indenting the message according to the hour; for example, messages received at 3 PM will be indented three blocks. This is intended to group messages by arrival time, thus facilitating searches and browsing. If users knows the time at which a message arrived, they can identify the range of messages in which it must fall using the navigation bar.

For the weather forecasts, since the day of the week for a forecast is always the information to be encoded, Irwin can use a semantic encoding. Irwin generates a list of seven blocks representing Sunday through Saturday, highlighting the one corresponding to the day of the forecast (see Figure 3).

The navigation bar is used to control the visible portion of the textual list. A black line next to a series of messages indicates that they are visible in the textual list. A black dot points out the pattern for the currently selected message.

2.4 Textual views

Graphics alone might not show all the information necessary to understand a message – at some point a textual view will likely help. Irwin incorporates two textual views: a header list and a message view. Irwin constructs lists of headers for each resource, the senders for email and Usenet news, abbreviated headers for Web pages, and days of the week for the weather forecast. If the user selects an item from the list, the message view shows a more informative summary of the original message. The user can click on the message view to jump to a browser to see the full message.

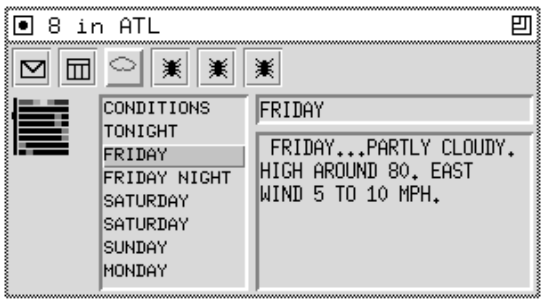

Figure 3: Irwin's display of the Atlanta weather forecast. The navigation bar contains a coded line for each day of the week - the colored dot shows which day is represented, with Sunday appearing at the far left. The message view provides detailed information about the upcoming forecast for the selected day.

2.5 Anticipated Irwin tasks

We designed Irwin with three potential tasks in mind, checking, browsing, and searching.

- **checking** - This task involves nothing more than examining Irwin when an auditory or visual change has occurred. Irwin provides quick access to the most recent changes during checking by listing the items in reverse chronological order with the newest item selected and on top of the list. We expect checking will be the most frequently performed task since it requires little physical and cognitive effort.
- **browsing** - In this task, the user scans through old messages looking for items of interest. We expect that users will notice repeated patterns in the navigation bar that indicate repeated messages with the same topic or author. We hope that certain patterns for desired topics will become familiar to users to the point where they can identify and select these items while browsing.
- **searching** - In this task, the user searches for a particular message, perhaps related to a recently received message. This task involves matching two or more patterns; for example, a selected pattern with one that occurs earlier in the list. We hope that users will perform searches in Irwin when they are reminded of a previous message by a newly arrived one.

3 Case Study

In our case study, we want to learn how people use technology to increase information awareness. What tasks do users want to do with an awareness tool? Do icons that change appearance provide useful information? Do auditory cues increase awareness in a positive way, or will they become intrusive or annoying over time? Are syntactic encodings usable and learnable? By offering Irwin to users and observing how they use it, we hope to answer some of these questions.

3.1 Methodology

Nine people attended a presentation and demonstration explaining how to set up and use Irwin. Seven people tried Irwin at least once, and four of those continued to use it regularly.

We asked those who did not use Irwin why they chose not to. Most simply did not feel like they needed such a tool. One person noted that she didn't receive much email and only rarely browsed the Web: "I'm an example of a fish who doesn't want a bicycle for Christmas - I just don't have much use for it!" One user wanted a less passive visualization tool. He reported the he felt he did not have enough control over when the resources were checked. "At times I want to tell it to go get information about a newsgroup or Web page. I don't want to just wait for the regular time interval checks."

Five months after the presentation we visited with each of the four frequent users for about an hour to discover how they used Irwin, and in particular which features were most and least useful. Many of our questions were open-ended, allowing the users to expand upon issues of interest. We also asked them to discuss other possible uses for awareness and monitoring tools. Their responses are summarized in the next section.

3.2 Individual reports

The four participants in this study are all members of the Georgia Tech Graphics, Visualization, and Usability Center. This allows us to assume a certain level of sophistication with email, Usenet news, the Web, and interfaces in general. Alan is a faculty member with a private office and his own Sun workstation, while the other three are graduate students who use X-terminals in cubicles and workstations in a shared laboratory.

The participants' real names have been changed to preserve their anonymity, and none of the participants were required to use Irwin or to answer any questions about it. We hope these assurances helped to generate a realistic experience.

3.2.1 Participant 1 - Alan

Alan uses Irwin to monitor his email, the Golf Magazine GolfOnline Web site, ESPN television's ESPNET Web sports news wire, and the local Atlanta weather. He places Irwin in the lower right corner of the screen with all views always visible (see Figure 4).

Alan uses a different auditory cue for each resource. "I chose nice short sounds that are moderately loud – long ones get old." Alan relies primarily on the auditory cues to inform him of changes; he infrequently looks to see if the icon appearances have changed and sometimes even has the icon view partially covered with other windows.

Alan receives around sixty email messages a day, and he always leaves his email tool running. When he receives new mail, he almost always turns to his email tool rather than to Irwin. Alan mainly uses Irwin for passive browsing of Web sites that he normally would not visit very often. Since Web sites change at irregular intervals, Irwin alerts Alan of changes without requiring him to check the site himself. Alan typically looks only at the currently displayed message when it pops up and rarely clicks on the icons or the navigation bar to view other resources or earlier messages. Sometimes he browses through the textual list and selects messages to view on the message display. When he sees a particularly interesting message, he uses Irwin to pull up the full article in his Netscape Web browser.

Figure 4: Alan's screen. Note the positioning of Irwin in the lower right corner. Alan is also showing an image in a Web page that he would like to monitor with Irwin.

Since Alan does not use Irwin to browse old email or Web summaries, he doubts he would use any navigation tool, whether it be a syntactic or semantic encoding or even a regular scrollbar. The only desired feature would be a thread display for email that would highlight threads of related messages, a feature not present in his mail reader. This added value might be useful, but as it is he prefers to pull up his mail reader when he needs to do any searching or browsing.

Alan wants Irwin to monitor other information resources, in particular Unix commands for checking the status of shared hardware resources. Sometimes when he prints a large document, he needs a tool to monitor the print queue and alert him when the job is done. Alan noted that monitoring the status of processes might be useful to see if they become defunct or exit abruptly. He would be willing to do a one-time configuration if it were then easy to start and stop these type of monitors. In addition, there are certain Web sites with images that Alan would like to monitor; for example, traffic report maps and weather radar pictures. He would like for Irwin to display a miniaturized version of these images.

3.2.2 Participant 2 - Bert

Bert uses Irwin to monitor the local weather and six different email folders. Since Bert filters his email, he needs a tool like Irwin to help keep track of changes in each folder. Bert configured Irwin so that only the icons are visible unless the cursor is inside the Irwin window. Bert places his Irwin at the upper left corner of his screen above to his clock and system load monitor (see Figure 5).

Bert has the same auditory cue (a beep) for all of his email folders. When he hears it he looks at the Irwin icons to see which folder received email. "It's less trouble just to look at Irwin than to remember lots of different sounds." Bert likes the fact that highlighted icons fade over time because it reminds him that there is email that he has been neglecting. He likes the changing weather icon but is unsure about having other icons change appearance. "Just so it's simple and makes sense."

Bert uses Irwin primarily to alert him when new email arrives. He likes the quick access

Figure 5: Bert's screen with Irwin in the upper right corner. Note that only the icons view of Irwin is visible. To see the other views, Bert clicks on one of the icons.

to information that Irwin provides. "It's much quicker than starting email, so I use only Irwin when possible." Bert pulls up his mail reader when he needs to read long messages or reply to a message, functions that are beyond the scope of Irwin. Often times he knows from the message summary that he can ignore the email until a more convenient time.

Bert generally does not look at the navigation bar, though sometimes frequently repeated patterns will catch his eye. He never tries to do any searching or browsing with Irwin, opting instead to use his email reader and Web browser for such tasks. He might use the navigation bar more if the information provided were more useful, like perhaps showing the importance of the message in some way. He actually finds the weather navigation bar's encoding of the day of week more useful than the syntactic encoding.

Bert would like to configure Irwin to monitor people's activities, in particular to see if they are available. Currently, he must run the Unix finger command repeatedly to see when people are at their machines. This type of repetitive task would be well-suited for a tool like Irwin. Bert also wants his clock and system load monitoring tools integrated into Irwin to save room on his desktop. He notes that some operating systems provide these types of toolbars, but often they are not very configurable.

3.2.3 Participant 3 - Carl

Carl uses Irwin to monitor his email, the comp.parallel.pvm Usenet newsgroup, the ESP-NET news wire, and the local weather. He places Irwin at the top of the screen next to his clock, and he always leaves all of the Irwin views visible (see Figure 6).

The only sound Carl configured in Irwin was a beep for email. He listened to a few of the sounds but decided against using them. "They would drive me up the wall." Carl even complained that some of the longer and louder sounds from other people's Irwins would disturb him.

Instead of sounds, Carl relies on the icons for information about new messages. He particularly likes the weather icon and wishes that other icons changed appearance as well.

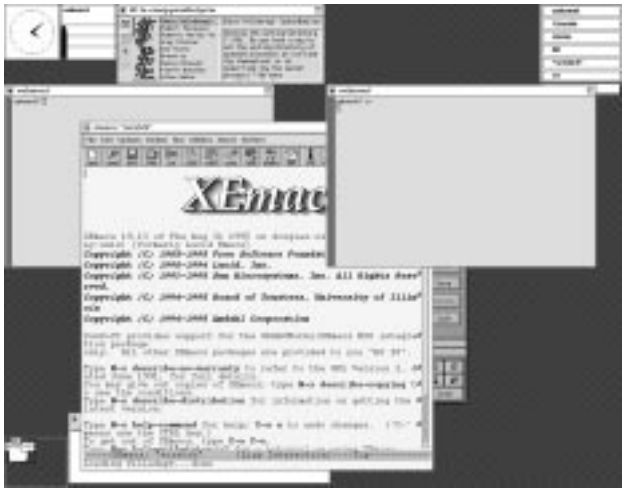

Figure 6: Carl's screen with Irwin in the upper left corner.

Since he often uses an X-terminal with a black-and-white monitor, neither the change in color of the icons nor the syntactic encoding of keywords are of much help. He would like to associate icons with different events so that when the event occurs the corresponding icon will be visible.

Carl tends to use Irwin for browsing more than the other users. However, he says he only recognizes two patterns in the navigation bar (one being his own encoded login name). Occasionally Carl will notice repeated patterns in some newsgroup representation; generally this means a knowledgeable person is replying to a number of messages at once, which makes for interesting reading.

Carl likes to read the Irwin textual message summaries for news articles but is frustrated that they are truncated after a certain number of words. Usually he wants to see the entire message without having to jump to his news reader. Similarly for Web pages, he does not want the slow-loading graphical view used by most browsers and would prefer a quick view of the text only. For Carl, the big advantage of Irwin is the speed at which he can obtain information.

Carl believes that a variety of views in the navigation bar would be more useful. At times he would prefer a site-of-origin view where the bars are colored by the site at which the message originated, and other times he wants a message threads view. "It should be easy to configure to see alternate views." In addition, Carl wants to be able to zoom in and out of the representation, or perhaps even have a fisheye view of the resource. He is concerned that he cannot see the entire resource at once.

3.2.4 Participant 4 - Dora

Dora uses Irwin to monitor her email, USA Today's Washington DC news wire, the local weather, and two Usenet newsgroups: comp.graphics.algorithms and comp.graphics.animation. She places Irwin at the bottom of the screen between her system performance and desktop management tools (see Figure 7).

Dora associates a different sound for each category of resources; for example, Irwin barks

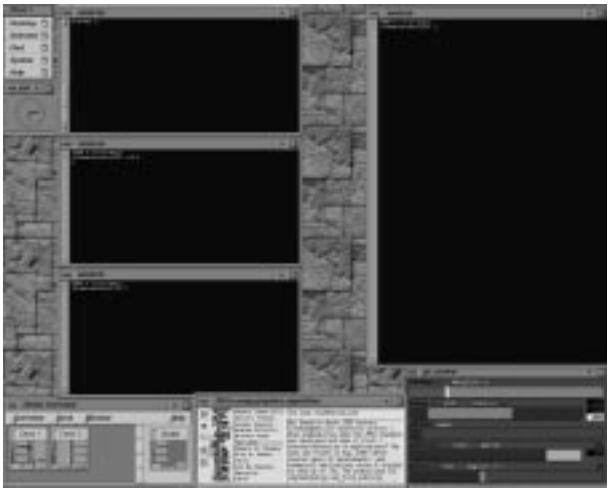

Figure 7: Dora's screen with Irwin at the bottom. Note that Dora configured Irwin to be larger than the Irwins for the other participants.

when new email arrives and beeps for any new news. She would prefer to combine related resources into a single icon. Then she would know at a single glance whether a resource of immediate interest has experienced changes. "At busy times I might check my email but not my newsgroups. It would be helpful to only have to check a single icon."

Dora is the only participant who uses semantic highlighting. She created three lists of names for her email: one for coworkers, a second for family members, and a third for her PhD committee members. Each list of names has a different highlight color, so when she looks at the navigation bar she can tell who has been sending her email. "When I see a lot of red, that generally means I'm in trouble with somebody." Dora still leaves the syntactic encoding on as well so she can distinguish between other messages. However, she does not recognize any names, though she sometimes notices repeated patterns resulting from several messages from the same person.

Since she generally works in a lab or cubicle without windows, Dora enjoys the changing icon of the weather display. "It serves as a reminder of what's going on in the real world." She would like to see something similar for other resources because it could provide easily accessible information - the user would only have to look at Irwin learn about the state of the resources.

4 Lessons Learned

Based on our observations of Irwin users, we would like to discuss a few of the lessons we learned which should help in the future development of information awareness tools.

Our users were willing to sacrifice a little bit of space to increase their awareness of information resources. Even those who did not use Irwin cited reasons other than space limitations. Most users wanted to be able to integrate other monitoring tools into Irwin as well, which bodes well for the window managers and desktop environments that tend to use this all-in-one philosophy for their toolbars.

To assist in knowing which resource changed, a few users created their own "auditory

hierarchy" by assigning related resources the same sound, and they indicated that they wanted to create a similar hierarchy with the icons. In such a situation, icons would be pull-down menus instead of buttons, and the user could create menus based on importance. When any item in the menu changes, the icon would change. Such a technique would save space, a top priority when dealing with information awareness tools.

When using Irwin, the primary task performed by our users was the checking of changed resources. Most users utilized some combination of feedback from auditory cues and icon changes to know when to check a resource. Though we provided auditory cues that were up to two seconds in duration, all of the sounds configured by our users were under one second.

Some of our users stated that they would notice repeated patterns in the syntactic encodings when browsing resources. However, none of the users could associate more than a few encodings with the corresponding word or name, and none tried to do any searches using the encodings. Why did the users not learn the encodings over time? Perhaps because they never saw the text and its encoding together and could not build the mental associations, or perhaps because the encodings are too complex and difficult to differentiate. It might simply be that the users did not take the time to look at the patterns because they always used other tools for browsing and searching. Both of these tasks benefit from a large amount of screen space, which is not practical in awareness tools like Irwin.

Too often visualizations will provide complex graphical views of a data space but will make it difficult to see the actual data that is represented. Ideally, a visualization will provide a view that combines graphics and text dependent on the available space, the needs of the user, and the amount of information that needs to be shown. If one of these factors changes, the way in which the information should be shown may change as well.

5 Acknowledgements

Thanks to all of the participants who used Irwin and gave their comments. Thanks also to John Stasko and Don Allison for reading previous drafts of this paper. Special thanks to Tom Rowan and the ORNL VIS Group for their support in the development of Irwin.

References

[BNG89] Megan L. Brown, Sandra L. Newsome, and Ephraim P. Glinert. An experiment into the use of auditory cues to reduce visual workload. In *Proceedings of the ACM SIGCHI '89 Conference on Human Factors in Computing Systems*, pages 339–346, Austin, TX, May 1989.

[Bre94] Steven A. Brewster. *Providing a structured method for integrating non-speech audio into human-computer interfaces.* PhD thesis, University of York, UK, 1994.

[BSM91] Ronald M. Baecker, Ian Small, and Richard Mander. Bringing icons to life. In *Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems*, pages 1–6, New Orleans, LA, May 1991.

[ESS92] Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner Jr. SeeSoft—a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, November 1992.

[FD92] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, December 1992.

[Fur86] George W. Furnas. Generalized fisheye views. In *Proceedings of the ACM SIGCHI '86 Conference on Human Factors in Computing Systems*, pages 16–23, Boston, MA, April 1986.

[HH94] William C. Hill and James D. Hollan. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10(2), April 1994.

[MA96] D. Scott McCrickard and Gregory D. Abowd. Assessing the impact of changes at the architectural level: A case study on graphical debuggers. In *Proceedings of the International Conference on Software Maintenance*, Monterey, CA, November 1996.

[MR96] D. Scott McCrickard and Thomas H. Rowan. Monitoring and visualizing information resources. Technical Report ORNL/TM-13193, Oak Ridge National Laboratory, July 1996.

[YGM95] Tak W. Yan and Hector Garcia-Molina. Sift - a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, 1995.